S.No: 1	Exp. Name: ***Project Module***	Date:2024-06-05

Aim:

Project Module

Source Code:

hello.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>

#define MAX_BOOKS 100
#define MAX_TITLE_LENGTH 100
#define MAX_AUTHOR_LENGTH 100
#define MAX_CATEGORY_LENGTH 50

// Structure to represent a book
typedef struct {
    char title[MAX_TITLE_LENGTH];
    char author[MAX_AUTHOR_LENGTH];
    char category[MAX_CATEGORY_LENGTH];
    bool is_available;
    bool is_borrowed;
    int borrowed_date;
    int due_date;
} Book;

Book library[MAX_BOOKS];
int num_books = 0;

// Function prototypes
void addBook();
void displayBooks();
void borrowBook();
void returnBook();
void calculateFine(int book_index, int return_date);
void searchBook();

int main() {
    int choice;

    do {
        printf("\nLibrary Management System\n");
        printf("1. Add Book\n");
        printf("2. Display Books\n");
        printf("3. Borrow Book\n");
        printf("4. Return Book\n");
        printf("5. Search Book\n");
        printf("6. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
```

```
        switch(choice) {
            case 1:
                addBook();
                break;
            case 2:
                displayBooks();
                break;
            case 3:
                borrowBook();
                break;
            case 4:
                returnBook();
                break;
            case 5:
                searchBook();
                break;
            case 6:
                printf("Exiting...\n");
                break;
            default:
                printf("Invalid choice! Please enter a number between 1 and
6.\n");
        }
    } while(choice != 6);

    return 0;
}

void addBook() {
    if (num_books >= MAX_BOOKS) {
        printf("Cannot add more books. Library is full.\n");
        return;
    }

    Book new_book;
    printf("Enter Title: ");
    scanf(" %[^\n]s", new_book.title);
    printf("Enter Author: ");
    scanf(" %[^\n]s", new_book.author);
    printf("Enter Category: ");
    scanf(" %[^\n]s", new_book.category);
    new_book.is_available = true;
    new_book.is_borrowed = false;
    library[num_books++] = new_book;
    printf("Book added successfully.\n");
}

void displayBooks() {
    if (num_books == 0) {
        printf("No books in the library.\n");
        return;
    }

    printf("Books in the library:\n");
    printf("Title\t\tAuthor\t\tCategory\tAvailable\n");
    for (int i = 0; i < num_books; i++) {
```

```
        printf("%s\t\t%s\t\t%s\t\t%s\n", library[i].title, library[i].author,
library[i].category, library[i].is_available ? "Yes" : "No");
    }
}

void borrowBook() {
    if (num_books == 0) {
        printf("No books in the library.\n");
        return;
    }

    char search_title[MAX_TITLE_LENGTH];
    printf("Enter the title of the book you want to borrow: ");
    scanf(" %[^\n]s", search_title);

    for (int i = 0; i < num_books; i++) {
        if (strcmp(library[i].title, search_title) == 0) {
            if (library[i].is_available) {
                library[i].is_available = false;
                library[i].is_borrowed = true;
                printf("Book '%s' borrowed successfully.\n", library[i].titl
e);
                printf("Enter borrowed date (YYYYMMDD): ");
                scanf("%d", &library[i].borrowed_date);
                printf("Enter due date (YYYYMMDD): ");
                scanf("%d", &library[i].due_date);
                return;
            } else {
                printf("Sorry, the book '%s' is currently not available for bo
rrowing.\n", library[i].title);
                return;
            }
        }
    }

    printf("Book with title '%s' not found.\n", search_title);
}

void returnBook() {
    if (num_books == 0) {
        printf("No books in the library.\n");
        return;
    }

    char search_title[MAX_TITLE_LENGTH];
    printf("Enter the title of the book you want to return: ");
    scanf(" %[^\n]s", search_title);

    for (int i = 0; i < num_books; i++) {
        if (strcmp(library[i].title, search_title) == 0 && library[i].is_borro
wed) {
            library[i].is_available = true;
            library[i].is_borrowed = false;
            int return_date;
            printf("Enter return date (YYYYMMDD): ");
            scanf("%d", &return_date);
```

```
            calculateFine(i, return_date);
            printf("Book '%s' returned successfully.\n", library[i].title);
            return;
        }
    }

    printf("Book with title '%s' not found or not borrowed.\n", search_title);
}

void calculateFine(int book_index, int return_date) {
    int due_date = library[book_index].due_date;
    if (return_date > due_date) {
        int days_overdue = return_date - due_date;
        int fine = days_overdue * 5; // Assuming a fine of $5 per day
        printf("Fine for overdue: $%d\n", fine);
    } else {
        printf("No fine. Book returned on time.\n");
    }
}

void searchBook() {
    if (num_books == 0) {
        printf("No books in the library.\n");
        return;
    }

    char search_term[MAX_TITLE_LENGTH];
    printf("Enter title or author to search: ");
    scanf(" %[^\n]s", search_term);

    printf("Search Results:\n");
    printf("Title\t\tAuthor\t\tCategory\tAvailable\n");
    for (int i = 0; i < num_books; i++) {
        if (strstr(library[i].title, search_term) != NULL || strstr(library
[i].author, search_term) != NULL) {
            printf("%s\t\t%s\t\t%s\t\t%s\n", library[i].title, library[i].auth
or, library[i].category, library[i].is_available ? "Yes" : "No");
        }
    }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Hello World